AR-008-815

DSTO-TR-0150

Test-Bed Performance Analysis of the Fiber Distributed Data Interface

Alan Allwright

19960212 254

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Test-Bed Performance Analysis of the Fiber Distributed Data Interface

Alan Allwright

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-TR-0150

ABSTRACT

A network analyser developed as part of the Distributed Processing task, NAV87/226.3, has been used to measure the media access delays on a Fiber Distributed Data Interface (FDDI) network. This report presents the results obtained in a series of experiments designed to test the utility of the network analyser.

APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury. South Australia, Australia, 5108

Telephone: (08) 259 7085
Fax: (08) 259 5980

AR-008-815
Feb 1995

Test-Bed Performance Analysis of the Fiber Distributed Data Interface

EXECUTIVE SUMMARY

This report describes the results obtained from a series of experiments to test a Fiber Distributed Data interface network analyser. The network analyser was developed within ITD. Each of the experiments is discussed and the results are related to the operation of the Fiber Distributed Data Interface protocol and the capabilities of the network analyser.

THIS PAGE INTENTIONALLY BLANK

Authors

Alan M. Allwright
Information Technology Division

Alan Allwright graduated from the South Australian Institute of Technology in 1988. Between 1990 and 1994 Alan worked on a Masters degree in computing. His thesis titled "Performance Analysis of Distributed Databases for Combat Systems" was submitted in 1995. Alan now works on computer simulation and System Engineering related tasks within C3ISE group in ITD.

THIS PAGE INTENTIONALLY BLANK

CONTENTS

Figures

Tables

Appendices

ABBREVIATIONS

AMD	Advanced Micro Devices
B_O	Buffer empty probability
B_{0P1}	High Priority Buffer Empty Probability
B_{0P2}	Low Priority Buffer Empty Probability
B_1	Buffer Full probability
B_{0P1}	High Priority Buffer Full Probability
B_{0P2}	Low Priority Buffer Full Probability
FDDI	Fiber Distributed Data Interface
IAT	Inter Arrival Times
ISO	International Standards Organisation
km	Kilometre
MAC	Media Access Controller
Mbps	Megabits per second
Mhz	Megahertz
N_{serv}	Total Number of Requests Serviced
N_{tot}	Total Number of Requests
OSI	Open Systems Interconnection
P_{async}	Probability of asynchronous request loss
P_{loss}	Probability of request loss
P_{serv}	Probability of request service
P1	Asynchronous High Priority Message
P2	Asynchronous Low Priority Message
PC	Personal Computer
T_OPR	Operative TTRT
T_PRI1	Asynchronous High Priority variable
T_PRI2	Asynchronous Low Priority variable
TTRT	Target Token Rotation Time
$Util_{sync}$	Synchronous Network Utilisation
$Util_{async}$	Asynchronous Network Utilisation
ms	Millisecond
ns	Nanosecond

THIS PAGE INTENTIONALLY BLANK

1 Introduction

The results obtained from a series of experiments designed to test the capabilities of a Fiber Distributed Data Interface (FDDI) network analyser (Ref. 1) are presented in this report. The development of the analyser was prompted by an investigation into the operation and performance of the FDDI network carried out as part of the Distributed Processing Task (NAV 87/226.3). A component of this investigation included the development of an FDDI simulation model (Ref. 2). The FDDI model simulates the operation of the FDDI Media Access Controller (MAC) and physical (PHY) layer protocols. The PHY and MAC layers are layers one and two of the International Standards Organisation (ISO) Open Systems Interconnection (OSI) 7-layer model for communication.

The network analyser was developed to measure the MAC and PHY transmission delays for FDDI frames on an FDDI network test-bed (Ref. 3). The measured delays are to be used to validate the simulation model. The analyser provides the user with facilities to specify network parameters such as the required Target Token Rotation Time (TTRT), the ring latency, and operational parameters such as frame lengths, inter-arrival times (IATs) and frame priorities.

Three experiments are presented. The first experiment investigates the influence of frame arrival rates and priorities on transmission delays. The effect of ring latency on transmission delays is examined in the second experiment. The third experiment measures the influence of the network "operative TTRT" (T_OPR) on transmission delays.

Section 2 discribes the network configuration for the experiments. The experiments and the experimental results are presented in sections 3, 4 and 5. Detailed results for each experiment are tabled in Appendices I, II and III respectively.

2 Background

The network for the experiments was configured as a single ring of three stations (Figure 1). A four channel data logger (Ref. 1), capable of logging all the network traffic, was attached to one station in a way which allowed frames generated by any of the three stations to be monitored. Each station was assigned a unique key to allow the logger to discriminate between the three traffic types, synchronous (key='A'), asynchronous high priority (key='B') and asynchronous low priority (key='C'). The stations inserted the keys into the frames before each frame was transmitted.

Each FDDI node comprised a 20 Mhz IBM Compatible PC, an AMD FDDI Fast card (a proprietary FDDI communications card), and a timer/interrupt card. An FDDI Delay Unit (Ref. 1) was placed between each station to allow the ring latency to be adjusted rather than being fixed for the network configuration. This allowed varying lengths of fibre optic cable to be simulated. A global time-base with a resolution of 0.256 ms was maintained by the data logger and distributed as an interrupt to each of the FDDI nodes. See "master clock" in Figure 1. Each FDDI node serviced its interrupts independently.

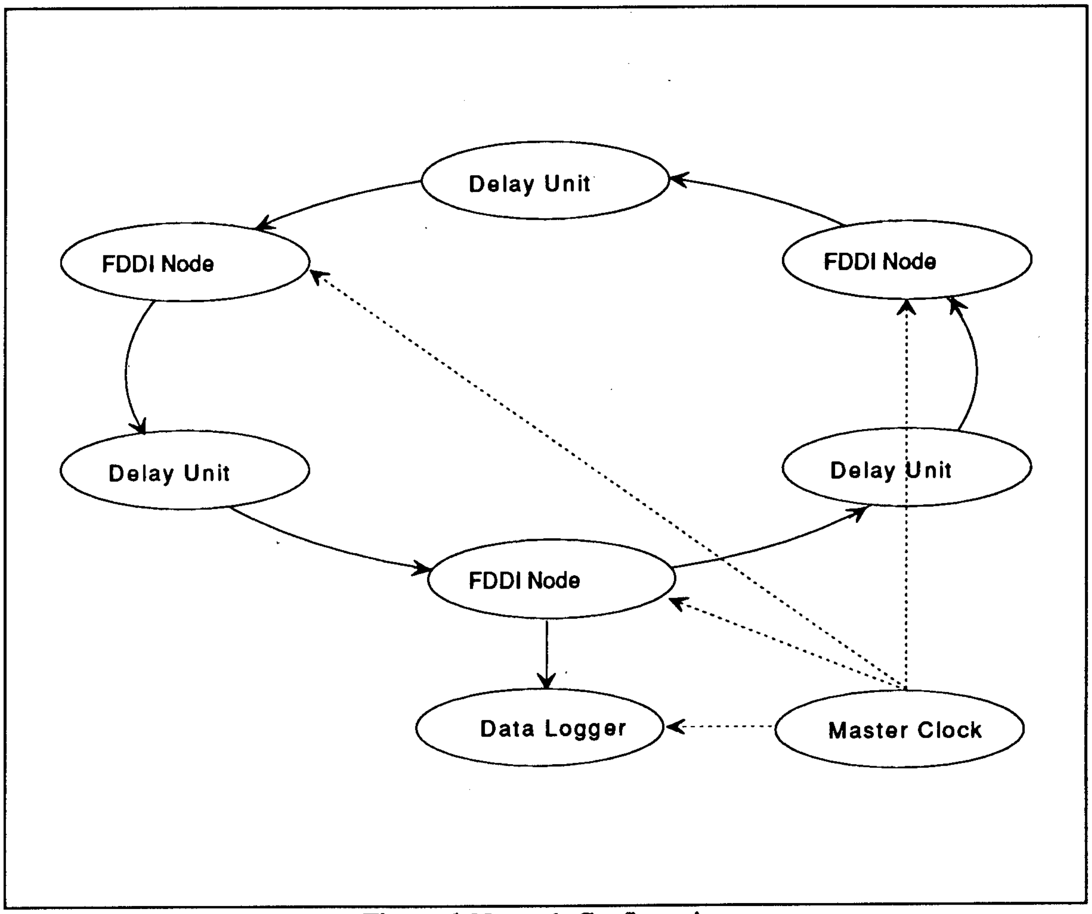

Figure 1 Network Configuration

Each experiment was divided into a number of 'runs' where each run required setting the network and station parameters, and running and logging transmission delays for a specific arrival rate of transmit requests. For example, the first run comprised setting the transmit request rate at 10 Mbps for each station and collecting sufficient frame delays from each station. The individual experimental parameters were set at the beginning of each run. The PDEMO program, supplied with the AMD Fast cards (Ref. 4), uses the requested TTRT to determine the operative TTRT (T_OPR). Individual station parameters (*eg* frame priority (T_PRI) and frame length) were set at the beginning of each run.

Once a transmit request is made by the node processor, the request is buffered in the FDDI card. The FDDI MAC protocol uses a timed token rotation protocol to manage access to the media. Once the token has been received, any waiting synchronous requests are serviced by copying the frame data from buffer memory onto the physical medium. Once all synchronous requests have been serviced, the hardware checks for asynchronous requests. Asynchronous requests will only be serviced if there is sufficient unutilised bandwidth during the current token rotation. Providing sufficient bandwidth is available with respect to the frame's priority threshold, the asynchronous request is serviced.

The following sections (Sections 3,4, and 5) discuss the experiments. The first experiment was conducted to test the FDDI nodes and the data logger with frame request rates from 10 to 100 Mbps. The second experiment was a repeat of experiment one except the ring latency was extended to 0.494 ms. The third experiment was conducted to investigate the effects of using a larger T_OPR.

3 Experiment 1

Experiment 1 was designed to test the FDDI nodes and the data logger, whilst introducing a minimum extra delay (minimum delay introduced by delay units) and using the minimum T_OPR. This experiment essentially tested the analyser's capability to load the network to 100 Mbps and for the logger to continually sample network traffic.

The frame data is pre-loaded into buffer memory in the FDDI cards. Frames stored in buffer memory are grouped into 'chains'. Grouping frames into chains simplifies the transmission of frames by allowing multiple frames to be transmitted by a single transmit request. Asynchronous frame priorities (P1, P2) were provided (Ref. 2) to allow comparisons with transmission delays produced with the simulation model.

The stations transmit at rates between 10 and 100 Mbps; the chain length, frame length and frame request IATs were set to make the request arrival rates exactly 10 Mbps intervals at each station. See Appendix I for chain, frame and interarrival-times. T_OPR was set to 4 ms (see Ref. 6 - minimum T_OPR), high frame priority (P1) was set to 0.000256 ms (high priority) and low frame priority (P2) was set to 3.372 ms (low priority). The results for this experiment are presented in Figures 2 and 3.

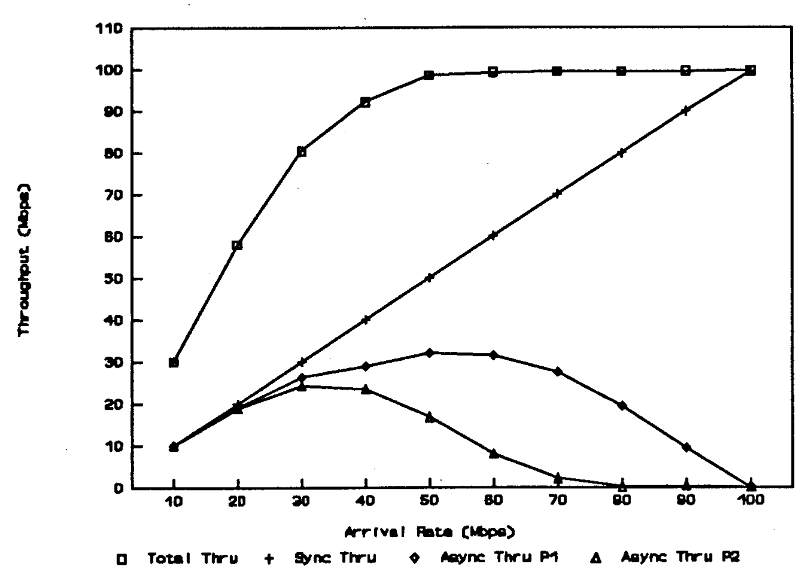

Figure 2 Experiment 1 - Throughput

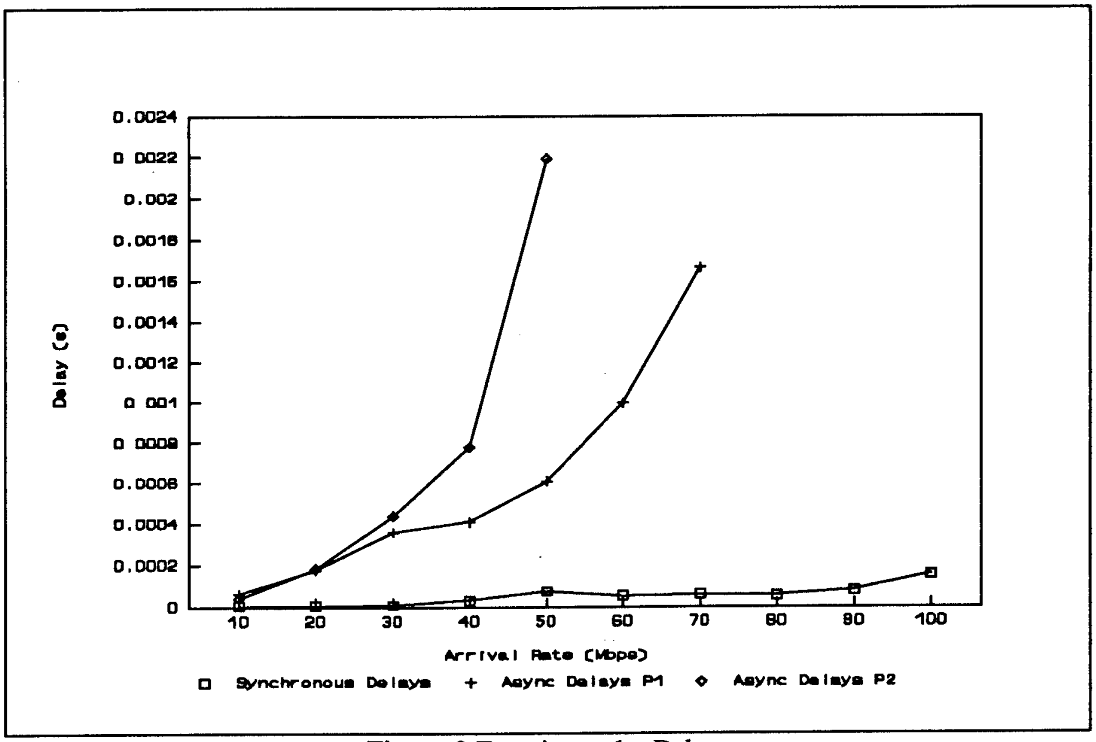

Figure 3 Experiment 1 - Delays

Minor deviations or "kinks" in the trends, for example the asynchronous P1 delays at 30 Mbps may be due to statistical sampling errors. The precise explanation for these deviations would require more detailed investigation of the circumstances contributing to the overall results. This type of investigation has not been done and is not discussed in this document.

The overall trends in these results can be explained in terms of the request loss probability, where requests are lost due to buffer overflows. Because the buffer capacity at each node is only two requests (Ref. 5), the system quickly reaches steady-state behaviour and request losses have an immediate effect on the throughput. Request loss probabilities are calculated by comparing the number of transmit requests made to the number of requests that are actually serviced. A count of the number of requests made is maintained by each node. A count of the number of requests serviced, for each node, is maintained by the data logger. The request loss probability (P_{loss}) is then calculated as one minus the service probability (P_{serv}), which is the number of requests serviced (N_{serv}) divided by the total number of requests made by each node processor (N_{tot}).

$$\begin{aligned} P_{loss} &= 1 - P_{serv} \\ &= 1 - N_{serv} \;/\; N_{tot} \end{aligned}$$

Request loss probabilities are shown in Table 1 and Figure 4. The synchronous and asynchronous request loss probabilities can be accounted for by considering the cyclic properties of the synchronous and asynchronous request rates which are described below.

Table 1 Experiment 1 Request Loss Probabilities (P_{loss})

Request Rate (Mbps)	10	20	30	40	50	60	70	80	90	100
Sync	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Async P1	0.00	0.04	0.13	0.10	0.20	0.34	0.53	0.51	0.77	1.00
Async P2	0.00	0.06	0.19	0.27	0.58	0.83	0.96	0.99	0.99	1.00

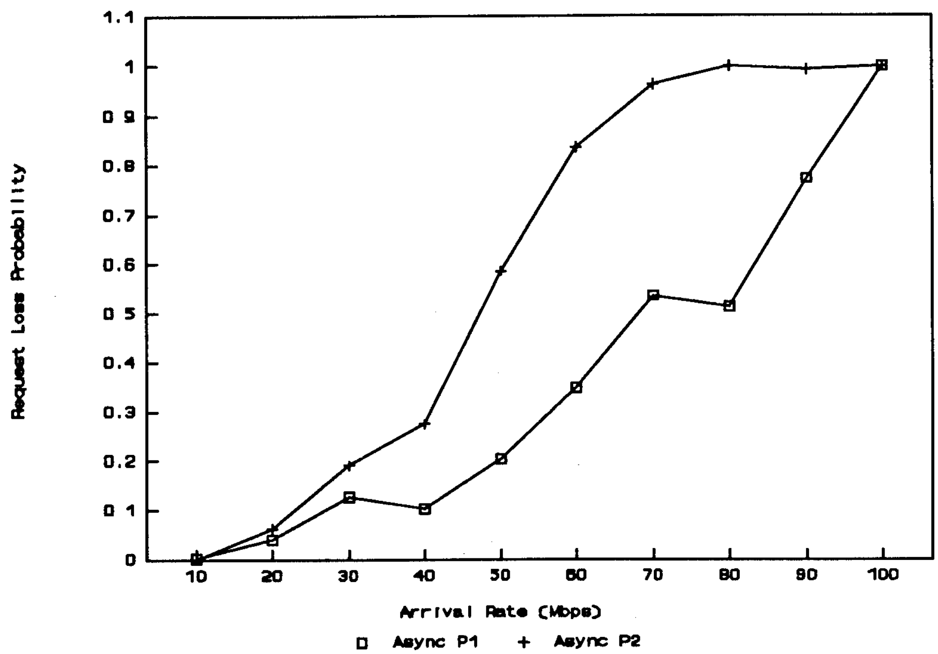

Figure 4 Experiment 1 - Request Loss Probabilities

The experiments have been designed such that synchronous requests occur with a long period and relatively large bursts. Asynchronous requests occur in short bursts with a high frequency. For example, 30 Mbps synchronous requests of 23040 bytes are made every 6.144 ms; asynchronous requests at the same data rate are for 1920 bytes every 0.512 ms. Figure 4 graphs three distinct trends for asynchronous P1 and P2 data between 10 to 40 Mbps, 40 to 70 Mbps and 70 to 100 Mbps.

3.1 10 to 40 Mbps

At 10 Mbps there is sufficient spare bandwidth that all the synchronous and asynchronous requests are serviced. Losses at this level occur due to the random variation of IAT's. IAT's for asynchronous requests are selected from a uniform distribution within the range 0.256 ms to 2*(expected(IAT)-0.5). If a sequence of three or more requests occur at 0.256 ms then buffer losses will occur. Because the probability of a sequence of short IATs is low, the overall probability of a request loss is small.

Between 10 and 33 Mbps there is enough bandwidth to fully service the three stations, and the load is generally small enough that none of the priority thresholds becomes significant. The request loss probabilities in this case result from asynchronous request losses that occur when asynchronous requests are made whilst synchronous requests are being serviced.

The following calculations provide an indication of the significance of this effect at 30 Mbps (note the measured results are bracketed [] for comparison):

Synchronous IAT = 24 ticks = 6.144 ms,
Frequency = 163 / sec,
Each chain of 10, 2304 Byte synchronous requests require approximately 1.8 ms to service (10 frames * 2304 bytes * 8 bits / 100 Mbps).

Average asynchronous IAT = 2 ticks = 0.512 ms,
Frequency = 1953 / sec.

While the synchronous request is being serviced an average of 1.8/0.512 = 3.5 asynchronous requests occur. The buffer is either empty or has one request resident. There will therefore be between 1 and 3 requests lost.

By introducing the probability that the buffer is either empty (B_0) or has one request resident (B_1), it is possible to calculate the request loss probability at 30 Mbps for the two asynchronous priority classes.

<u>Priority 1 (P1)</u>

The request loss probability at any instant is dependant upon whether or not a synchronous request is being made at the time the asynchronous request is being made.

The request loss probability (P_{loss}) is the sum of the request loss probability whilst synchronous services are occurring (P_{serv}) plus the request loss probability when synchronous services are not occurring (P_{ns}).

$$P_{loss} = P_{serv} + P_{ns}$$

The aim of this calculation is to show the effect of synchronous service times on the request loss probability for high priority asynchronous requests. Consequently the request loss probability when synchronous requests are not occurring (P_{ns}) is ignored.

The probability P_{serv} is dependant upon two components; the synchronous utilisation of the network ($Util_{sync}$) and the probability an asynchronous request is lost (P_{async}). $Util_{sync}$ is calculated as the proportion of time per second spent servicing synchronous requests.

$$P_{sync} = Util_{sync} * P_{async}$$

The proportion of time spent per second servicing the synchronous request ($Util_{sync}$) is calculated by using the synchronous service time (as above 1.8 ms) and the synchronous request frequency (163 / sec) :

$$Util_{sync} = 1.8 \text{ ms} * 163 = 0.29$$

The probability an asynchronous request is lost (P_{async}) can be calculated by enumerating all the possible asynchronous arrival sequences during a synchronous service and calculating the relative probability a request is lost. An approximation to this follows, where the relative probabilities are calculated only for the case where 3 or 4 arrivals occur during the synchronous service time; it is also assumed that either 3 or 4 arrivals will occur with equal probability. These calculations are done by taking into account the expected request arrival rate and the expected buffer utilisation measured by the probabilities B_0 and B_1.

If the buffer is empty (B_0) and 3 requests arrive, during the synchronous service, 1 request is lost. If the buffer is empty and 4 requests arrive, 2 requests are lost. If 3 or 4 requests arrive with equal probability, on average 1.5 ((1+2)/2) asynchronous requests are lost per synchronous service. Since the average arrival rate is 3.5 requests per synchronous service, probability an asynchronous request is lost (B_{0P1}) is 1.5 / 3.5 = 0.43.

If the buffer has one asynchronous request resident (B_1) and 3 requests arrive, during the synchronous service, 2 requests are lost. If the buffer has one request resident and 4 requests arrive, 3 requests are lost. If 3 or 4 requests arrive with equal probability, on average 2.5 ((2+3)/2) asynchronous requests are lost per synchronous service. Since the average arrival rate is 3.5 requests per synchronous service, probability an asynchronous request is lost (B_{1P1}) is 2.5 / 3.5 = 0.71.

The buffer probabilities obtained from the data logger are:

$$B_0 = 0.67,\ B_1 = 0.21$$

The net request loss probability (P_{loss}) is calculated as the probability no requests are buffered (B_0) and subsequent requests are lost (B_{0P1}) plus the probability one request (B_1) is buffered and a subsequent request is lost (B_{1P1}) whilst synchronous service is occurring :

$$
\begin{aligned}
P_{loss} &= Util_{sync} * P_{async} \\
&= Util_{sync} * (B_0 * B_{0P1} + B_1 * B_{1P1}) \\
&= 0.29 * (0.67 * 0.43 + 0.21 * 0.71) \\
&= 0.13 \quad [0.13]
\end{aligned}
$$

Priority 2 (P2)

The number of asynchronous P2 requests that occur whilst the synchronous requests are being serviced, as calculated above must be increased to account for the asynchronous P1 requests that are serviced after the completion of the synchronous service. Immediately after synchronous service there will always be two asynchronous P1 requests waiting. These requests will take a further 0.3 ms (1920 bytes * bits * 2 frames / 100 Mbps) to service. The total service time, for synchronous and asynchronous requests is 2.1 ms (1.8 ms + 0.3 ms). During this service time, on average 2.1/0.512 = 4.1 asynchronous P2 requests will arrive. This results in a subsequent loss of asynchronous P2 requests whilst the asynchronous P1 requests are being serviced. Therefore buffer losses for P2 will be between 2 and 4 requests.

If the buffer is empty (B_0) and 4 requests arrive, during the synchronous service, 2 requests are lost. If the buffer is empty and 5 requests arrive, 3 requests are lost. If 4 or 5 requests arrive with equal probability, on average 2.5 ((2+3)/2) asynchronous requests are lost per synchronous service. Since the average arrival rate is 4.1 requests per synchronous service, the probability of an asynchronous request being lost (B_{0P2}) is 2.5 / 4.1 = 0.61.

If the buffer has one asynchronous request resident (B_1) and 4 requests arrive, during the synchronous service, 3 requests are lost. If the buffer has one request resident and 5 requests arrive, 4 requests are lost. If 4 or 5 requests arrive with equal probability, on average 3.5 ((3+4)/2) asynchronous requests are lost per synchronous service. Since the average arrival rate is 4.1 requests per synchronous service, probability of an asynchronous request being lost (B_{1P2}) is 3.5 / 4.1 = 0.85.

The buffer probabilities obtained from the data logger are:

$$B_0 = 0.68, \; B_1 = 0.13$$

The variable $Util_{sync}$ in this case must be increased to take into account not only the proportion of time spent servicing synchronous requests, but also the proportion of time spent servicing asynchronous P1 requests.

$$Util_{sync} = Util_{sync} + Util_{async}$$

The proportion of time spent per second servicing the asynchronous requests ($Util_{async}$) is calculated by using the asynchronous service time 0.3 ms. The synchronous request frequency (163 / second) is used in this case because this calculation determines the effect of asynchronous P1 service that occurs after synchronous service :

$$Util_{async} = 0.0003 * 163 = 0.05$$

$$Util_{sync} = Util_{sync} + Util_{async}$$

$$Util_{sync} = 0.29 + 0.05 = 0.34$$

The net request loss probability (P_{loss}) is calculated as the probability no requests are buffered (B_0) and subsequent requests are lost (B_{0P2}) plus the probability one request (B_1) is buffered and a subsequent request is lost (B_{1P2}) whilst synchronous service is occurring :

$$\begin{aligned} P_{loss} &= Util_{sync} * P_{async} \\ &= Util_{sync} * (B_0 * B_{0P2} + B_1 * B_{1P2}) \\ &= 0.34 * (0.68 * 0.61 + 0.13 * 0.85) \\ &= 0.18 \quad [0.19] \end{aligned}$$

While these calculations are somewhat simplified, they are close to the measured results, giving some confidence that the major cause for the buffer losses at 30 Mbps is contention for network bandwidth whilst synchronous requests are being serviced.

3.2 40 to 70 Mbps

Between 40 and 70 Mbps the previous effects are further compounded by the influence of the T_PRI thresholds.

At between 60 to 70 Mbps the service time for asynchronous P1 requests exceeds the priority threshold for P2 requests. This, in conjunction with asynchronous P1 and P2 requests having the same request rates, precludes any further service to asynchronous P2 requests and the request loss probability approaches unity.

3.3 70 to 100 Mbps

The drop in request loss probability experienced by P1 requests between 70 and 80 Mbps results from P2 requests no longer providing any competition and because P1 has such a high priority. This becomes equivalent to the 10 to 30 Mbps case with buffer overflows resulting largely from contention with synchronous services.

3.4 Maximum Synchronous Delays

The synchronous delays are capped by the timed token protocol. At 100 Mbps very few asynchronous requests are being serviced; at most one will be serviced each time the token is captured before the P1 threshold becomes significant. The maximum synchronous delays can be calculated as the sum of the waiting times for each request in the batch. Delays for synchronous requests are measured as the difference between the service times for each request in the batch and the request time for the batch.

The synchronous station will always service all its synchronous requests. Therefore the maximum synchronous delay is the sum of the service times for the two batches (of 10 frames), not including the last request, plus the time to service one asynchronous P1 requests.

```
Maximum synchronous delay

     = ((2 Batches (10 Frames)-1 Frame) * 3840 Bytes * 8 Bits / 100 Mbps)
     + ( 1 Frame * 3200 Bytes * 8 Bits / 100 Mbps )

     = 5.836 ms + 0.256 ms
     = 6.092 ms
```

This compares well with the maximum synchronous delay of 6.170 ms measured by the analyser.

3.5 Synchronous Buffer Overflow

In a well dimensioned network, synchronous request losses due to buffer overflow are not expected to occur. However, the data logger recorded synchronous buffer overflows at 100 Mbps.

The tabulated value for the synchronous request loss probability is 0.0003. Synchronous request losses at this request rate were due to cumulative delays in the processing of batch requests in the FDDI cards. It was found that, in the processing of batches, the FDDI cards introduced delays of less than 0.0005 ms (the resolution of the data logger clock) in the interframe latency within each batch.

Because request rates were calculated without taking these delays into account the request rate was set at one batch every 3.072 ms. The calculated service time for the batches is 3.072 ms, thus resulting in a required throughput of 100 Mbps. Due to this extra delay, the actual service time is 3.072 ms/batch + 9*0.0005 ms (interframe latency) = 3.0765 ms per batch. This extra service time resulted in a long term buffer overflow probability of less than 0.00146 (*viz* 1 - (3.072/3.0765)) at 100 Mbps.

4 Experiment 2

The objective of this experiment was to measure the effect of network latency on the performance of the network. The effect of increasing network latency is to steal transmit time away from the stations, which results in a reduction in the available bandwidth. Ulm, in reference 7, provides a good discussion of the effects of increased network latency.

All the network and station parameters are the same as for Experiment 1. The parameters for this experiment are detailed in Appendix II, the results are presented in Figures 5, 6 and 7, and request loss probabilities are presented in Table 2.

A special purpose delay unit (Ref. 1) is used to simulate the effects of longer network fibres, by delaying the transit of frames through the unit. The required delays are set at the beginning of the experiment by setting switches on the delay unit.

The network delays are symmetric in the network; each link introduces a delay of 0.164ms. This delay was selected so as to have a significant effect on the low priority threshold (0.628 ms). The delay is equivalent to a network of fibre and passive stations with approximately 32 km of fibre (at 0.005085 ms/km) or a total ring length of 96 km.

```
Network delays are calculated using the function :

Total Delay = no_stations*(2^(n-1)*switchable_delay+card_delay)+fixed_delay

no_stations = 3, switch setting n=10, switchable_delay = 320 ns
card_delay = 980 ns, fixed_delay = 5 ns

Total_Delay = 3 * (2^9 * 320 ns + 980) + 5
            = 0.494 ms    (approx 0.164 ms per station)
```

The T_OPR for this experiment is 4.0 ms. The introduced latency of 0.494 ms accounts for 0.494 / 4.0 = 12.35% of the operational time. Therefore the latency absorbs 12.25 % * 100 Mbps = 12.350 Mbps. Thus resulting in a maximum achievable throughput of 100 - 12.35 = 87.65 Mbps. This theoretical calculations correlate well with the measured results of a maximum measured throughput of 87.38 Mbps.

Table 2 Experiment 2 Request Loss Probabilities

Request Rate (Mbps)	10	20	30	40	50	60	70	80	90	100
Sync	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Async P1	0.00	0.01	0.22	0.24	0.35	0.51	0.74	0.82	0.99	0.99
Async P2	0.08	0.60	0.88	0.97	0.99	0.99	0.99	0.99	0.99	0.99

The throughput trends in this case can be explained in the same way as Experiment 1, except in this case ring latency has been increased to such an extent that asynchronous low priority (P2) throughput is immediately influenced by its priority threshold.

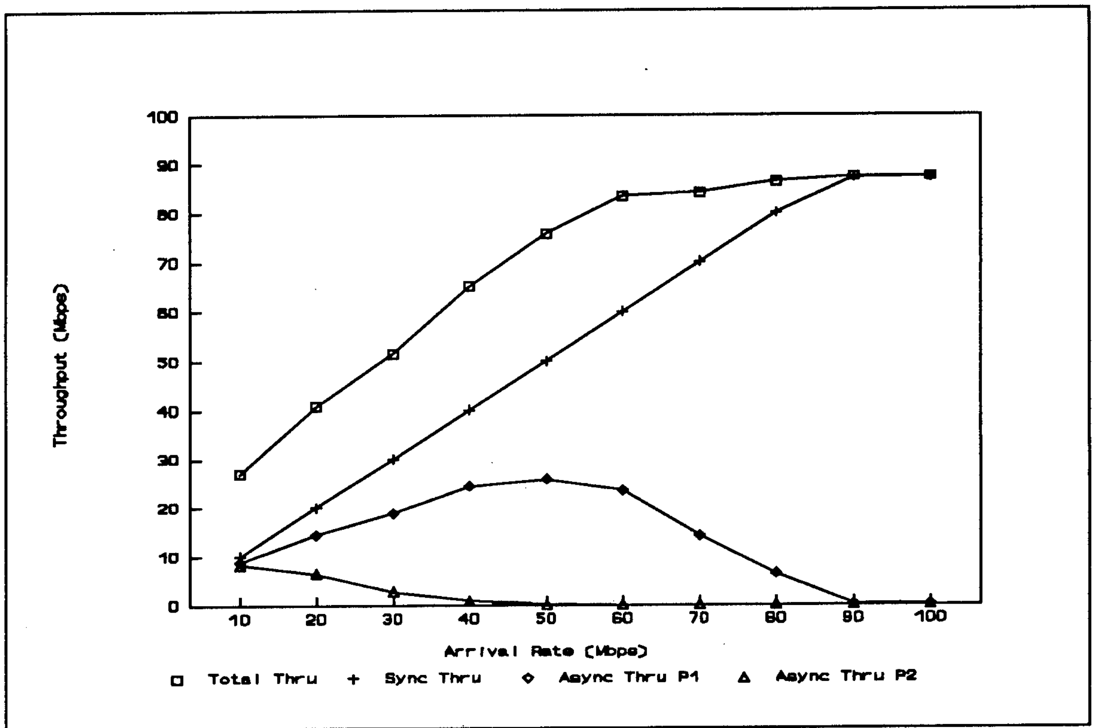

Figure 5 Experiment 2 - Throughput

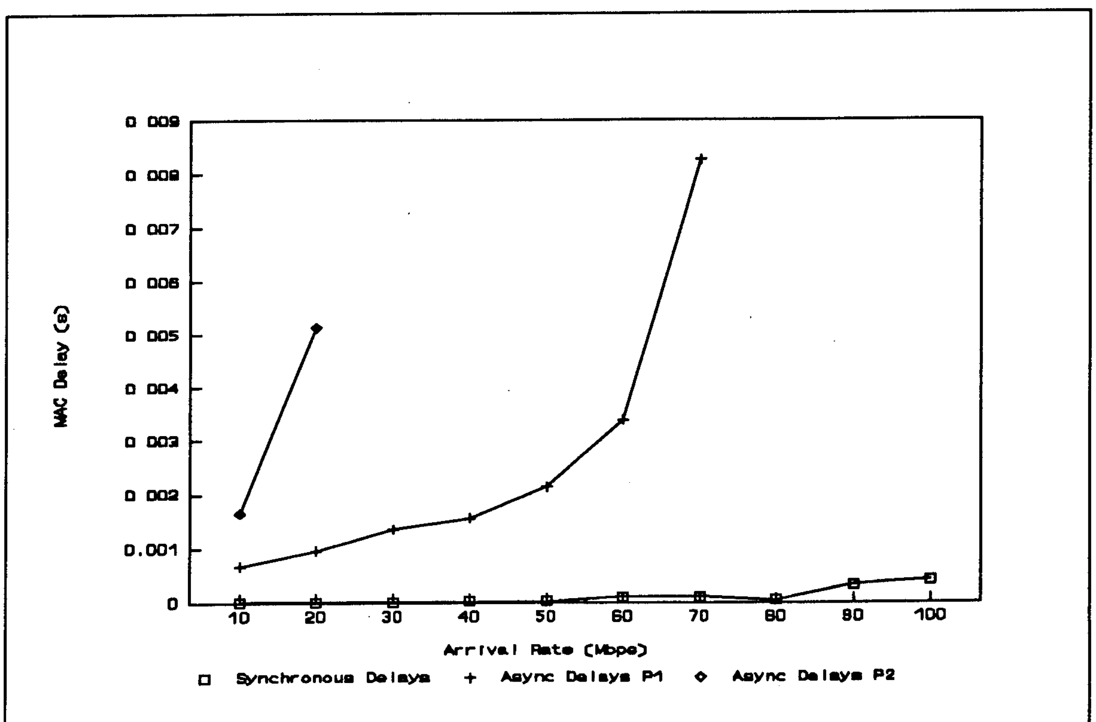

Figure 6 Experiment 2 - Delays

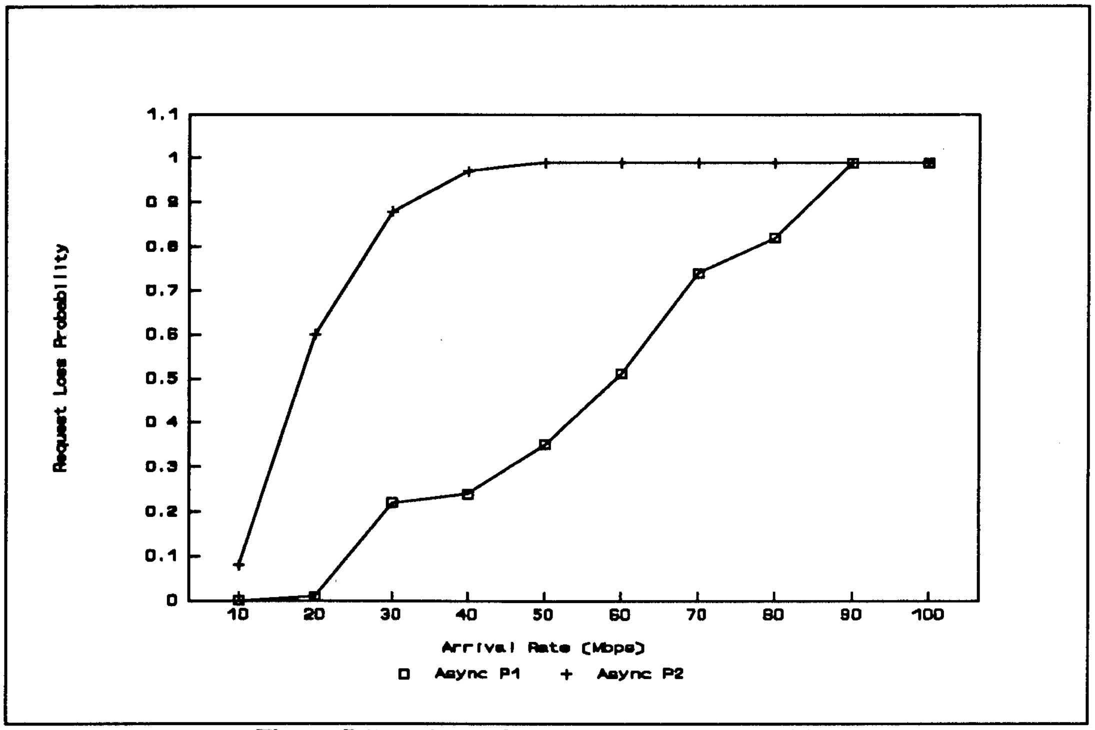

Figure 7 Experiment 2 - Request Loss Probabilities

On token rotations where no other requests have been serviced, 0.494 ms is lost to ring latency. The T_PRI1 threshold is 3.372 ms, T_OPR is 4 ms, which leaves only 0.134 ms (4.000 ms - 3.372 ms - 0.494 ms) available for frames to be transmitted. This level is almost immediately significant, where service times for asynchronous P1 requests start at 0.102 ms at 10 Mbps and increase to 256 ms at 50 Mbps. The general trend here is that asynchronous P1 requests between 10-40 Mbps are lost through contention with synchronous requests. Between 50-100 Mbps requests are lost through no available bandwidth.

5 Experiment 3

This experiment was designed to test the effect of using a larger T_OPR (24 ms). The larger T_OPR is used at the expense of the maximum achievable throughput before buffer overflows occur. By using a larger T_OPR it is possible to make synchronous requests with a much larger batch size than is possible using the minimum T_OPR (4.0 ms). Bux et. al. (Ref. 8) discuss the effects of T_OPR on the performance of an FDDI network. The experimental parameters and results are detailed in Appendix III. The batch length, frame length and IAT parameters are set to make calculations easier in this case and do not give the exact request rates.

To minimise the effect of buffer overflows associated with the larger T_OPR it is necessary to reduce the frequency at which requests occur. As a result of the reduced request frequency it is also necessary to use a larger batch size. Figures 8 and 9, respectively, show the measured throughput and request delays. Table 3 and Figure 10 show the measured request loss probabilities.

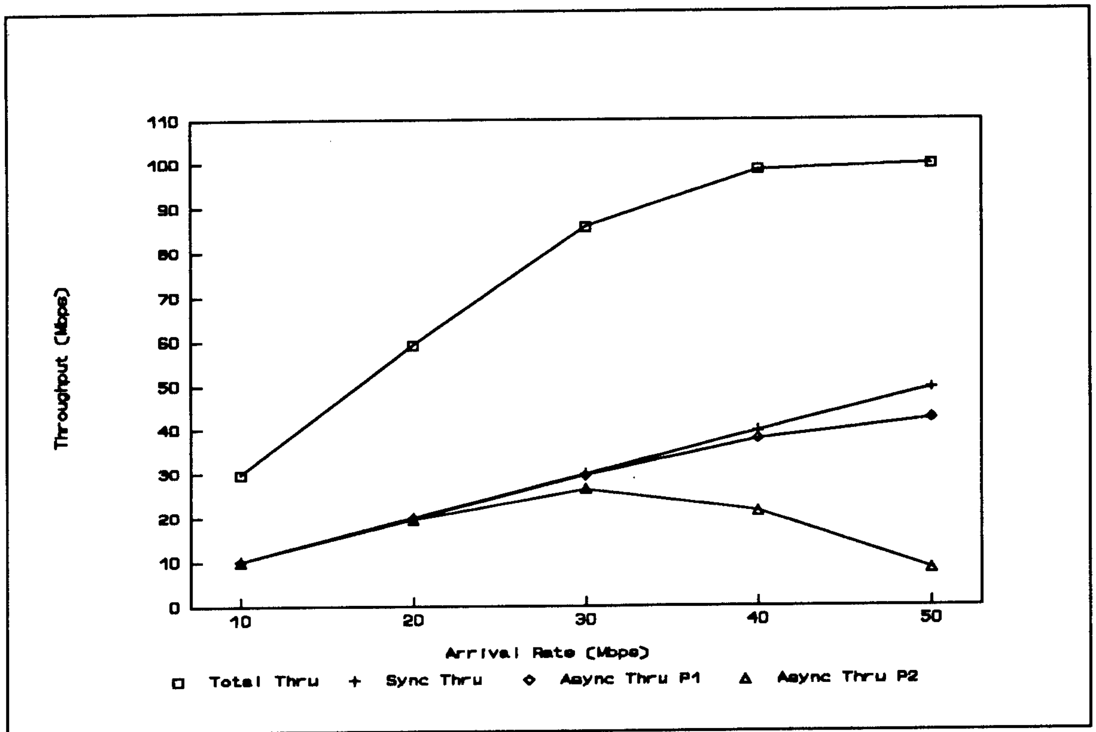

Figure 8 Experiment 3 - Throughput

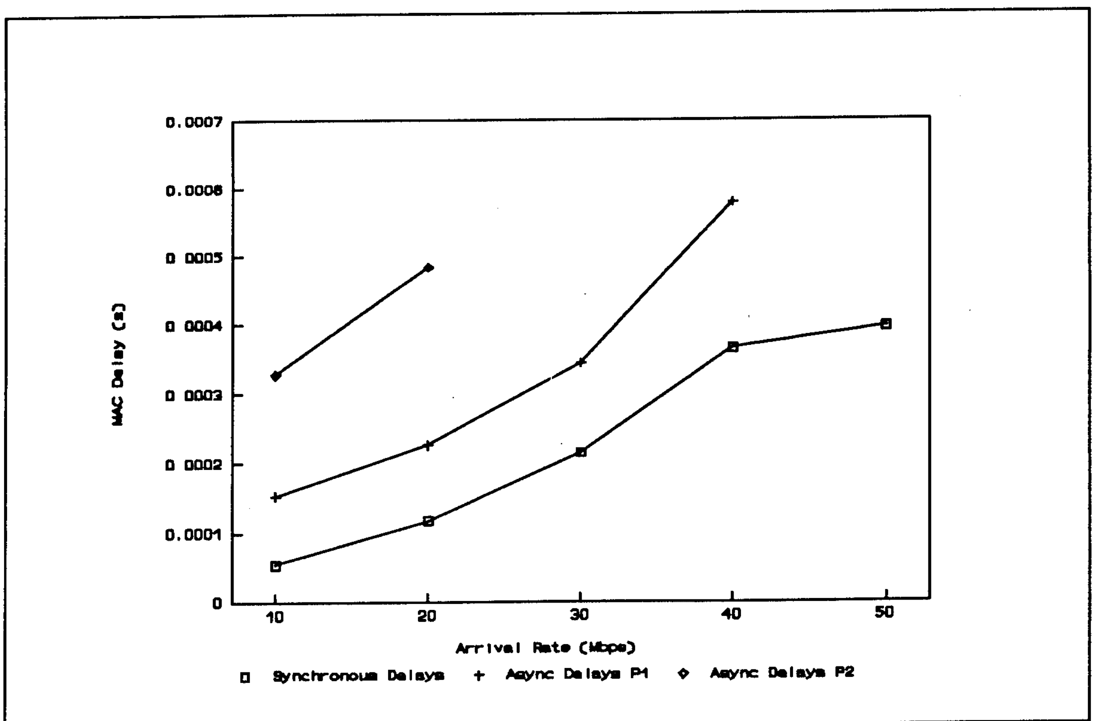

Figure 9 Experiment 3 - Delays

The results in this case reflect firstly, the much greater spread in asynchronous IAT's and secondly, the much higher priority given to asynchronous high priority (P1) requests. The higher spread in IAT's result from the reduced request frequency for asynchronous requests; this also results in lower request losses due to conflicts. The percentage for T_PRI2 against T_OPR is the same; because T_OPR is much greater T_PRI2 is much longer and there is significantly less chance of the priority threshold becoming significant. Overall this has resulted in greater P2 throughput up to 30 Mbps and consequently, when all stations are requesting at 30 Mbps, the overall throughput is closer to 90 Mbps than that experienced in the previous experiments.

Table 3 Experiment 3 Request Loss Probabilities

Request Rate (Mbps)	10	20	30	40	50
Sync	0.00	0.00	0.00	0.00	0.00
Async P1	0.00	0.00	0.00	0.00	0.01
Async P2	0.00	0.00	0.01	0.04	0.20

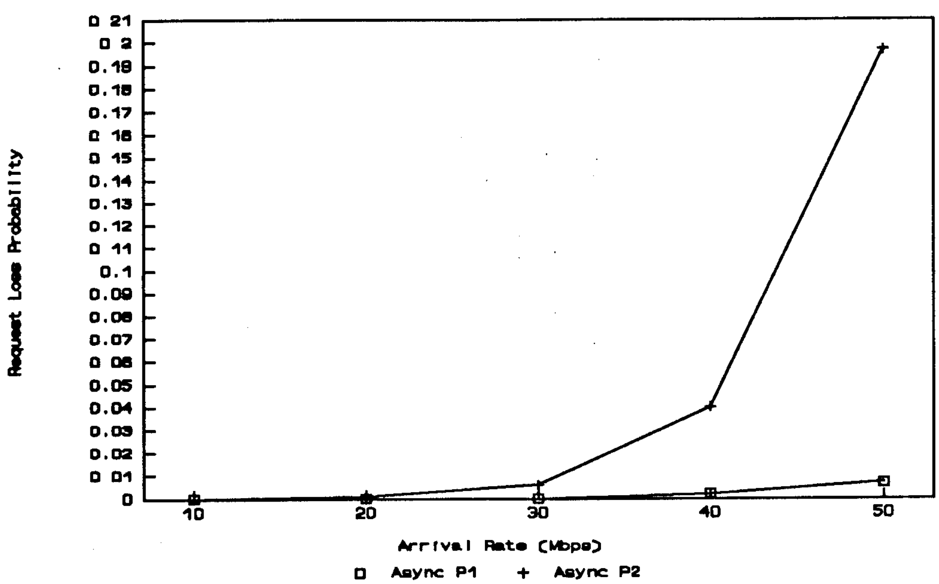

Figure 10 Experiment 3 - Request Loss Probabilities

6 Conclusion

The results from the experiments have been shown to correlate well with both published results and the known operation of the FDDI MAC and Physical levels. The analysis also shows the results to be strongly constrained by the available buffers for synchronous and asynchronous requests. Overall the experiments have shown the network analyser provides accurate and reliable results and as such is a useful tool in determining the MAC and Physical level delays in a variety of circumstances.

7 Acknowledgments

The author wishes to acknowledge with gratitude the assistance provided by Mr J.G. Schapel (ERL), Mr Reg Driver (ERL), and Mr Alan Wood (ERL) in the design and running of the experiments presented in this document.

8 References

No.	Author	Title
1	Allwright, A.M. Driver, R. Wood, A.	"FDDI Network Analyser" ERL-0695-RN, Draft February 1995
2	Scholz, M.L.	"Simulation of the FDDI Network: A Progress Report" CSI Working Paper 90/01, 1990
3	Miller, S.J.	"Distributed Processing Test-Bed System: First Interim Report for DPTBS", WSRL-TM26/90, September 1990
4	AMD	"SUPERNET Hardware and Software Support" Advanced Micro Devices, 1989
5	AMD	"THE SUPERNET Family for FDDI" Technical Manual, Advanced Micro Devices 1989
6	ANSI	"FDDI Token Ring Media Access Controller" X3.139-1987, 1986
7	Ulm, J.M.	"A Timed Token Ring Local Area Network and its Performance Characteristics" 7th IEEE Conference on Local Computer Networks, 1982
8	Bux, W. Dykeman, D.	"An investigation of the FDDI media-access control protocol" EFOC/LAN 87, pp 219-236, 1987

Appendix I

Experiment 1 Parameters and Results

This Appendix describes the parameters and results for Experiment 1. Each of the parameters is discussed and any calculations associated with these parameters are also provided.

Table I.1 Experiment 1 - Parameters

T_OPR = 4 ms   P1 = 0.00256 ms   P2 = 3.372 ms

Rate (Mbps)[1] Key	Chain[2] Length (Frames)	Frame[3] length (Bytes)	IAT[4] (0.256 ms Ticks)	Requests[5]	Serviced[6]	Arrival Generator[7]
10 A	5	1536	24	5000	5000	Fixed 24000 24
10 B	1	1280	4	5348	5341	Random 6000 4 1
10 C	1	1280	4	5347	5342	Random 6000 4 7000
20 A	10	1536	24	10000	10000	Fixed 24000 24
20 B	1	1280	2	9654	9264	Random 6000 2 1
20 C	1	1280	2	9657	9055	Random 6000 2 7000
30 A	10	2304	24	10000	10000	Fixed 24000 24
30 B	1	1920	2	9655	8435	Random 6000 2 1
30 C	1	1920	2	9658	7821	Random 6000 2 7000
40 A	5	3072	12	10000	10000	Fixed 24000 12
40 B	1	2560	2	9653	8664	Random 6000 2 1
40 C	1	2560	2	9656	7004	Random 6000 2 7000
50 A	5	3840	12	10000	10000	Fixed 24000 12
50 B	1	3200	2	9654	7679	Random 6000 2 1
50 C	1	3200	2	9658	4004	Random 6000 2 7000
60 A	10	2304	12	20000	20000	Fixed 24000 12
60 B	1	3840	2	9655	6291	Random 6000 2 1
60 C	1	3840	2	9658	1575	Random 6000 2 7000
70 A	10	2688	12	20000	20000	Fixed 24000 12
70 B	1	4480	2	9655	4698	Random 6000 2 1
70 C	1	4480	2	9658	358	Random 6000 2 7000
80 A	10	3072	12	20000	20000	Fixed 24000 12
80 B	2	2560	2	12576	5838	Random 6000 2 1
80 C	2	2560	2	9676	30	Random 6000 2 7000
90 A	10	3456	12	20000	20000	Fixed 24000 12
90 B	2	2880	2	10896	2480	Random 6000 2 1
90 C	2	2880	2	9072	86	Random 6000 2 7000
100 A	10	3840	12	19937	19930	Fixed 24000 12
100 B	2	3200	2	9668	16	Random 6000 2 1
100 C	2	3200	2	9667	4	Random 6000 2 7000

1. Rate (Mbps)/Key : This field specifies the data rate in Mbps and the frame identification key used by each of the stations.

2. Chain Length : The chain length specifies the number of frames to be transmitted upon each transmit request,

3. Frame Length : Length of the frame in bytes, including header (SA,DA etc).

4. IAT : The Inter-Arrival Time for requests at the transmitting station. The IAT is given in terms of 0.256 ms ticks. for eg. if the IAT = 24, transmit requests are made every 24 * 0.256 ms = 6.144 ms. Using the above three fields (2,3,4) it is possible to calculate the requested throughput, at 40 Mbps for example ;

Asynchronous P2

Chain length = 1, Frame Length = 2560 bytes = 20480 bits,
IAT = 2 Ticks = 0.512 ms

Request rate = Chain length * frame length (bits) / IAT (seconds)
= 1 * 20480 / 0.000512 seconds = 40 Mbps

5. Requests : This field is the number of requests made during the experiment.

6. Serviced : Is a count of the number of requests serviced, that is those requests which were logged at the data logger.

7. Arrival Generator : is the parameters used to generate the file of requests arrival times used by each transmitter.

a. For all synchronous requests (key='A'), the requests IAT are fixed, the first parameter specifies the tick count for the last request made and the second parameter specifies the IAT.

b. For Asynchronous requests the Turbo C++ (version 1.01) random number generator was used, the first parameter specifies the number of requests to be generated, the second parameter specifies the Mean IAT, and the third parameter specifies the random number seed to be used.

These fields can be used to calculate the run time for the experiment. For eg. Synchronous 10 Mbps the last request is made at 24000 ticks and the IAT for these requests is 24 ticks. Therefore 1000 Chain Synchronous requests will be generated over 24000 * 0.256 ms = 6.144 seconds.

For asynchronous 6000 requests were made with an IAT of 4 ticks. Asynchronous requests will be generated for 6000 * 4 * 0.256 ms = 6.144 seconds.

Table I.2 Experiment 1 - Results

Rate (Mbps)[1] Key	Delay[2] (500 ns Ticks)	Requests[3]	Serviced[4]	Frame[5] length (Bytes)	Service[6] Time (500ns Ticks)	Throughput[8] (Mbps)	Total[9] Throughput (Mbps)
10 A	10.04	5000	5000	1536	12288986	9.10	
10 B	123.31	5348	5341	1280	12291072	8.89	
10 C	82.24	5347	5342	1280	12289024	8.90	27.80
20 A	10.38	10000	10000	1536	12290223	19.10	
20 B	354.97	9654	9264	1280	12291072	15.44	
20 C	366.03	9657	9055	1280	12290560	15.09	50.52
30 A	18.43	10000	10000	2304	12291323	29.99	
30 B	716.90	9655	8435	1920	12292608	21.08	
30 C	873.98	9658	7821	1920	12292608	19.55	70.62
40 A	62.15	10000	10000	3072	12290037	39.99	
40 B	819.56	9653	8664	2560	12290048	28.88	
40 C	1554.91	9656	7004	2560	12290048	23.34	92.21
50 A	149.01	10000	10000	3840	12290594	49.99	
50 B	1217.26	9654	7679	3200	12291072	31.99	
50 C	4386.44	9658	4004	3200	12292608	16.68	98.65
60 A	102.04	20000	20000	2304	12291645	59.98	
60 B	1987.88	9655	6291	3840	12292608	31.44	
60 C	13913.20	9658	1575	3840	12292608	7.87	99.30
70 A	118.00	20000	20000	2688	12292232	69.98	
70 B	3323.53	9655	4698	4480	12292608	27.39	
70 C	66781.87	9658	358	4480	12292608	2.08	99.46
80 A	109.25	20000	20000	3072	12295155	79.95	
80 B	3098.74	12576	5838	2560	12295168	19.45	
80 C	660012.00	9676	30	2560	12295168	0.10	99.50
90 A	156.09	20000	20000	3456	12293100	89.96	
90 B	8672.23	10896	2480	2880	12293120	9.30	
90 C	222892.14	9072	86	2880	12293632	0.32	99.58
100 A	314.10	19937	19930	3840	12298786	99.56	
100 B	1536409.25	9668	16	3200	12299264	0.07	
100 C	6148441.00	9667	4	3200	12299776	0.02	99.65

1. Rate (Mbps)/key : As above.

2. Delay : The delays are those measured in the data logger for each of the traffic types. The delays presented here are in 500 ns ticks, the actual measured delay is for example 10 Mbps, key = 'A' ; 10.042800 * 0.0000005 ms = 0.00000502 ms.

3. Requests : As above.

4. Serviced : As above.

5. Frame Length : As above.

6. Service time : Total time over which traffic was being logged, in 500 ns clock ticks. The service times are obtained for each traffic class independently to isolate biases from initial and terminating conditions.

7. Tick Time : Data logger clock tick time = 500 ns.

8. Throughput : calculated throughput for each traffic type. The throughput is calculated using fields 4,5,6,7,8 for example the 40 Mbps synchronous (key='A') requests.

Serviced = 10000, Frame Length = 3072 Bytes, Service Time = 12290037 Ticks.

Throughput	=	bits per second
	=	Serviced * Frame Length * Bits / Service Time * Tick Time
	=	10000 * 3072 * 8 / 12290037 * 0.0000005
	=	245760000 / 6.140185
	=	39.9933702 Mbps

Throughput is calculated in this way to avoid and terminating and end condition biases.

9. Total Throughput : Total throughput is the sum of the three, (synchronous, asynchronous priority 1 and asynchronous priority 2), throughput for each run.

Appendix II

Experiment 2 Parameters and Results

This Appendix tables the parameters and results for Experiment 2. The meanings of the parameters and the experimental results are discussed in Appendix I.

Table II.1 Experiment 2 - Parameters

T_OPR = 4 ms P1 = 0.00256 ms P2 = 3.372 ms

Rate (Mbps) Key	Chain Length (Frames)	Frame length (Bytes)	IAT (0.256 ms Ticks)	Requests	Serviced	Arrival Generator
10 A	5	1536	24	5000	5000	Fixed 24000 24
10 B	1	1280	4	5348	5341	Random 6000 4 1
10 C	1	1280	4	5347	5342	Random 6000 4 7000
20 A	10	1536	24	10000	10000	Fixed 24000 24
20 B	1	1280	2	9654	9264	Random 6000 2 1
20 C	1	1280	2	9657	9055	Random 6000 2 7000
30 A	10	2304	24	10000	10000	Fixed 24000 24
30 B	1	1920	2	9655	8435	Random 6000 2 1
30 C	1	1920	2	9658	7821	Random 6000 2 7000
40 A	5	3072	12	10000	10000	Fixed 24000 12
40 B	1	2560	2	9653	8664	Random 6000 2 1
40 C	1	2560	2	9656	7004	Random 6000 2 7000
50 A	5	3840	12	10000	10000	Fixed 24000 12
50 B	1	3200	2	9654	7679	Random 6000 2 1
50 C	1	3200	2	9658	4004	Random 6000 2 7000
60 A	10	2304	12	20000	20000	Fixed 24000 12
60 B	1	3840	2	9655	6291	Random 6000 2 1
60 C	1	3840	2	9658	1575	Random 6000 2 7000
70 A	10	2688	12	20000	20000	Fixed 24000 12
70 B	1	4480	2	9655	4698	Random 6000 2 1
70 C	1	4480	2	9658	358	Random 6000 2 7000
80 A	10	3072	12	20000	20000	Fixed 24000 12
80 B	2	2560	2	12576	5838	Random 6000 2 1
80 C	2	2560	2	9676	30	Random 6000 2 7000
90 A	10	3456	12	20000	20000	Fixed 24000 12
90 B	2	2880	2	10896	2480	Random 6000 2 1
90 C	2	2880	2	9072	86	Random 6000 2 7000
100 A	10	3840	12	19937	19930	Fixed 24000 12
100 B	2	3200	2	9668	16	Random 6000 2 1
100 C	2	3200	2	9667	4	Random 6000 2 7000

Table II.2 Experiment 2 - Results

Rate (Mbps) Key	Delay (500 ns Ticks)	Requests	Serviced	Frame length (Bytes)	Service Time (500 ns Ticks)	Throughput (Mbps)	Total Throughput (Mbps)
10 A	5.03	5000	5000	1536	12288156	9.10	
10 B	658.15	5348	5304	1280	12291072	8.84	
10 C	1640.24	5346	4906	1280	12288512	8.18	27.03
20 A	8.09	10000	10000	1536	12290057	19.10	
20 B	951.34	9654	8601	1280	12291072	14.33	
20 C	5134.55	9657	3836	1280	12290560	6.39	40.72
30 A	12.58	10000	10000	2304	12290890	29.99	
30 B	1344.78	9654	7517	1920	12291072	18.79	
30 C	21559.80	9658	1083	1920	12292608	2.71	51.48
40 A	24.07	10000	10000	3072	12289409	39.99	
40 B	1550.99	9653	7285	2560	12290048	24.28	
40 C	103505.43	9655	235	2560	12289536	0.78	65.06
50 A	21.39	10000	10000	3840	12289845	49.99	
50 B	2141.08	9653	6184	3200	12290048	25.76	
50 C	3512938.25	9656	7	3200	12290048	0.03	75.78
60 A	92.70	20000	20000	2304	12293905	59.97	
60 B	3396.43	9657	4681	3840	12295168	23.39	
60 C	1231260.00	9660	2	3840	12294656	0.01	83.37
70 A	94.72	20000	20000	2688	12292552	69.97	
70 B	8264.60	9655	2419	4480	12292608	14.11	
70 C	1236736.00	9658	2	4480	12292608	0.01	84.10
80 A	43.84	20000	20000	3072	12291567	79.98	
80 B	10323.49	10604	1898	2560	12292608	6.32	
80 C	3101757.00	9660	4	2560	12292608	0.01	86.31
90 A	320.86	19460	19400	3456	12295152	87.25	
90 B	881233.81	9669	24	2880	12295168	0.09	
90 C	3100371.00	9663	4	2880	12295168	0.01	87.35
100 A	405.27	17722	17470	3840	12294214	87.31	
100 B	1755641.00	9664	14	3200	12295168	0.06	
100 C	6371018.00	9662	4	3200	12294656	0.02	87.38

Appendix III

Experiment 3 Parameters and Results

This Appendix tables the parameters and results for Experiment 3. The meanings of the parameters and the experimental results are discussed in Appendix I.

Table III.1 Experiment 3 - Parameters

T_OPR = 24 ms P1 = 0.002560 ms P2 = 12.645 ms

Rate (Mbps) Key	Chain Length (Frames)	Frame Length (Bytes)	IAT (0.256 ms Ticks)	Requests	Serviced	Arrival Generator
10 A	20	4500	282	20000	20000	Fixed 282000 282
10 B	20	4500	282	19563	19540	Random 1000 282 1
10 C	20	4500	282	19689	19680	Random 1000 282 2000
20 A	20	4500	141	20000	20000	Fixed 141000 141
20 B	20	4500	141	19867	19860	Random 1000 141 1
20 C	20	4500	141	19440	19420	Random 1000 141 2000
30 A	20	4500	94	20000	20000	Fixed 94000 94
30 B	20	4500	94	19694	19680	Random 1000 94 1
30 C	20	4500	94	17710	17600	Random 1000 94 2000
40 A	20	4500	71	20000	20000	Fixed 71000 71
40 B	20	4500	71	19141	19100	Random 1000 71 1
40 C	20	4500	71	11210	10760	Random 1000 71 2000
50 A	20	4500	57	20000	20000	Fixed 57000 57
50 B	20	4500	57	17296	17160	Random 1000 57 1
50 C	20	4500	57	4161	3340	Random 1000 57 2000

Table III.2 Experiment 3 - Results

Rate (Mbps) Key	Delay (500 ns Ticks)	Requests	Serviced	Frame Length (Bytes)	Service Time (500 ns Ticks)	Throughput (Mbps)	Total Throughput (Mbps)
10 A	53.91	20000	20000	4500	144397704	9.97	
10 B	152.86	19563	19540	4500	141800446	9.92	
10 C	327.48	19689	19680	4500	144427008	9.81	29.70
20 A	116.53	20000	20000	4500	72205714	19.94	
20 B	225.83	19867	19860	4500	72149585	19.82	
20 C	483.01	19440	19420	4500	72231936	19.35	59.12
30 A	214.68	20000	20000	4500	48149478	29.91	
30 B	343.76	19694	19680	4500	48156672	29.42	
30 C	713.82	17710	17600	4500	48150528	26.32	85.65
40 A	365.61	20000	20000	4500	36369276	39.59	
40 B	579.67	19141	19100	4500	36406784	37.77	
40 C	2165.00	11210	10760	4500	36427133	21.27	98.63
50 A	396.31	20000	20000	4500	29203100	49.31	
50 B	772.13	17296	17160	4500	29206016	42.30	
50 C	8121.47	4161	3340	4500	29220352	8.23	99.84

DISTRIBUTION

		No. of Copies
Defence Science and Technology Organisation		
Chief Defence Scientist	)	
Central Office Executive	)	1 shared copy
Counsellor, Defence Science, London		Cont Sht
Counsellor, Defence Science, Washington		Cont Sht
Senior Defence Scientific Adviser		1 copy
Scientific Adviser POLCOM		1 copy
Aeronautical & Maritime Research Laboratory		
Director, Aeronautical & Maritime Research Laboratory		1 copy
Chief Air Operations Division		Cont Sht
Chief Maritime Operations Division		Cont Sht
Chief Weapon Systems Division		Cont Sht
Navy Office		
Navy Scientific Adviser		1 copy
Army Office		
Scientific Adviser, Army		1 copy
Airforce Office		
Air Force Scientific Adviser		1 copy
Defence Intelligence Organisation		
Assistant Secretary Scientific Analysis		1 copy
Electronics & Surveillance Research Laboratory		
Chief Information Technology Division		1 copy
Chief Electronic Warfare Division		Cont Sht
Chief Communications Division		Cont Sht
Chief Land, Space and Optoelectronics Division		Cont Sht
Chief High Frequency Radar Division		Cont Sht
Chief Microwave Radar Division		Cont Sht
Research Leader Command & Control and Intelligence Systems		1 copy
Research Leader Military Computing Systems		1 copy
Manager Human Computer Interaction Laboratory		Cont Sht
Executive Officer, ITD		Cont Sht
Head Software Engineering Group		Cont Sht
Head, Trusted Computer Systems Group		Cont Sht
Head, Command Support Systems Group		Cont Sht
Head, Intelligence Systems Group		Cont Sht
Head, Systems Simulation and Assessment Group		Cont Sht
Head, Exercise Analysis Group		Cont Sht
Head, C3I Systems Engineering Group		1 Copy
Mr J. Schapel, Information Acquisition & Processing Group		1 Copy
Mr A. Allwright, C3I Systems Engineering Group		1 Copy
Mr D. O'Dea, Information Managment Group		1 Copy
Mr A. Wood, C3I Systems Engineering Group		1 Copy

Head, Computer Systems Architecture Group	Cont Sht
Head, Information Management Group	Cont Sht
Head, Information Acquisition & Processing Group	Cont Sht
Publications & Publicity Officer ITD	1 copy

Libraries and Information Services

Australian Government Publishing Service	1 copy
Defence Central Library, Technical Reports Centre	1 copy
Manager, Document Exchange Centre, (for retention)	1 copy
National Technical Information Service, United States	2 copies
Defence Research Information Centre, United Kingdom	2 copies
Director Scientific Information Services, Canada	1 copy
Ministry of Defence, New Zealand	1 copy
National Library of Australia	1 copy
Defence Science and Technology Organisation Salisbury, Research Library	2 copies
Library Defence Signals Directorate Canberra	1 copy
British Library Document Supply Centre	1 copy
Parliamentary Library of South Australia	1 copy
The State Library of South Australia	1 copy

Spares

Defence Science and Technology Organisation Salisbury, Research Library	6 copies

Department of Defence

DOCUMENT CONTROL DATA SHEET

1. Page Classification	UNCLASSIFIED
2. Privacy Marking/Caveat	N/A

3a. AR Number	3b. Establishment Number	3c. Type of Report	4. Task Number
AR-008-815	DSTO-TR-0150	TECHNICAL REPORT	87/226.3

5. Document Date	6. Cost Code	7. Security Classification	8. No. of Pages	36
FEBRUARY 1995	803885	U U U Document Title Abstract	9. No. of Refs.	8

10. Title
Test-Bed Performance Analysis of the Fiber Distributed Data Interface

S (Secret C (Confi) R (Rest) U (Unclass)
* For UNCLASSIFIED docs with a secondary distribution LIMITATION, use (L) in document box.

11. Author(s)
Alan Allwright

12. Downgrading/ Delimiting Instructions
N/A

13a. Corporate Author and Address
Information Technology Division
Electronics and Surveillance Research Laboratory
PO Box 1500 SALISBURY SA 5108

14. Officer/Position responsible for

Security N/A

Downgrading N/A

Approval for release CITD

13b. Task Sponsor
Navy

15. Secondary Release Statement of this Document

APPROVED FOR PUBLIC RELEASE.

Any enquiries outside stated limitations should be referred through DSTIC, Defence Information Services, Department of Defence, Anzac Park West, Canberra, ACT 2600.

16a. Deliberate Announcement

No limitation.

16b. Casual Announcement (for citation in other documents)

[√] No Limitation [ ] Ref. by Author, Doc No and date only

17. DEFTEST Descriptors
Fiber distributed Data Interface
Network analysers

18. DISCAT Subject Codes
NA

19. Abstract

A network analyser developed as part of the Distributed Processing task, NAV87/226.3, has been used to measure the media access delays on a Fiber Distributed Data Interface (FDDI) network. This report presents the results obtained in a series of experiments designed to test the utility of the network analyser.